

# A Decentralized Second-Order Method for Dynamic Optimization

Aryan Mokhtari, Wei Shi, Qing Ling, and Alejandro Ribeiro

**Abstract**—This paper considers decentralized dynamic optimization problems where nodes of a network try to minimize a sequence of time-varying objective functions in a real-time scheme. At each time slot, nodes have access to different summands of an instantaneous global objective function and they are allowed to exchange information only with their neighbors. This paper develops the application of the Exact Second-Order Method (ESOM) to solve the dynamic optimization problem in a decentralized manner. The proposed dynamic ESOM algorithm operates by primal descending and dual ascending on a quadratic approximation of an augmented Lagrangian of the instantaneous consensus optimization problem. The convergence analysis of dynamic ESOM indicates that a Lyapunov function of the sequence of primal and dual errors converges linearly to an error bound when the local functions are strongly convex and have Lipschitz continuous gradients. Numerical results demonstrate the claim that the sequence of iterates generated by the proposed method is able to track the sequence of optimal arguments.

**Index Terms**—multi-agent network, decentralized optimization, dynamic optimization, second-order methods

## I. INTRODUCTION

We consider a decentralized dynamic consensus optimization problem where the components of a *time-varying* global objective function are available at different nodes of a network. Specifically, consider a discrete time index  $t \in \mathbb{N}$ , a decision variable  $\tilde{\mathbf{x}} \in \mathbb{R}^p$ , and a connected network containing  $n$  nodes where each node  $i$  has access to a dynamic local objective  $f_{i,t} : \mathbb{R}^p \rightarrow \mathbb{R}$ . The agents' goal is to track the time-varying optimal argument

$$\tilde{\mathbf{x}}_t^* := \operatorname{argmin}_{\tilde{\mathbf{x}} \in \mathbb{R}^p} \sum_{i=1}^n f_{i,t}(\tilde{\mathbf{x}}), \quad (1)$$

while exchanging information with their neighbors only. Henceforth, we refer to  $f_{i,t}$  as the instantaneous local function of node  $i$  at time  $t$  and to  $\sum_{i=1}^n f_{i,t}$  as the instantaneous aggregate or global objective at time  $t$ . Distributed dynamic problems like the one in (1) are used to formulate problems in distributed signal processing [1]–[3], distributed control [4]–[6], and multi-agent robotics [7]–[9].

For the static version of (1) – with local functions  $f_{i,t} = f_i$  that are time invariant and, consequently, with a fixed global objective as well –, there exist numerous descent methods

that can solve the problem in a decentralized fashion. Some of these algorithms implement first order descent in the primal domain [10], [11], some others rely on first order ascent in the dual domain [12]–[16], and some recent efforts attempt to utilize second order information [17]. Since the dynamic problem in (1) can be interpreted as a sequence of static optimization problems, any of the methods in [10]–[17] can be used as a solution methodology. However, the methods are themselves iterative and their application would require running a large number of (inner) iterations for each of the (outer) time steps  $t$ ; see, e.g., [18].

Dynamic methods avoid the introduction of multiple time steps and consider that only a few steps of an iterative optimization method are executed for each time index  $t$  [3], [19]–[24]. Naturally, these methods track  $\tilde{\mathbf{x}}_t^*$  with some error because as they implement a descent on  $\sum_{i=1}^n f_{i,t}$ , the function drifts towards  $\sum_{i=1}^n f_{i,t+1}$ . These dynamic methods are therefore concerned with characterizing the tracking error [3], [19]–[24] and with developing specific techniques to reduce the steady state gap between the estimated and actual optima [23], [24]. Our goal in this paper is to develop the application of the recently proposed exact second order method (ESOM) [25] for solving the decentralized dynamic optimization problem in (1).

We begin by introducing decentralized equivalents of (1) (Section II) and propose the use of the dynamic ESOM method to solve the resulting decentralized dynamic optimization problem. Dynamic ESOM is a primal-dual algorithm that uses a quadratic approximation of an augmented Lagrangian (Section III). This approximation is expected to have good convergence properties because it incorporates second order information. Alas, this quadratic approximation requires access to the Hessian inverse of the augmented Lagrangian, which is not locally computable. This issue is resolved by using a truncation of the Taylor's series expansion of the Hessian inverse [17] (Section III-A). We study convergence properties of dynamic ESOM and show that the sequence of iterates it generates converges linearly to a neighborhood of the sequence of optimal arguments  $\tilde{\mathbf{x}}_t^*$  (Section IV). We perform a numerical evaluation of the performance of dynamic ESOM in solving a dynamic least squares problem (Section V) and close the paper with concluding remarks (Section VI).

**Notation.** Vectors are written as  $\mathbf{x} \in \mathbb{R}^p$  and matrices as  $\mathbf{A} \in \mathbb{R}^{p \times p}$ . Given  $n$  vectors  $\mathbf{x}_i$ , the vector  $\mathbf{x} = [\mathbf{x}_1; \dots; \mathbf{x}_n]$  represents a stacking of the elements of each individual  $\mathbf{x}_i$ . We use  $\|\mathbf{x}\|$  and  $\|\mathbf{A}\|$  to denote the Euclidean norm of vector  $\mathbf{x}$  and matrix  $\mathbf{A}$ , respectively. The norm of vector  $\mathbf{x}$  with

Work supported by NSF CAREER CCF-0952867, ONR N00014-12-1-0997, and NSFC 61004137. A. Mokhtari and A. Ribeiro are with the Dept. of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA. (aryanm, aribeiro@seas.upenn.edu). W. Shi is with the Coordinated Science Lab., University of Illinois at Urbana-Champaign, 1308 W Main St, Urbana, IL 61801, USA. (wilburs@illinois.edu). Q. Ling is with the Dept. of Automation, University of Science and Technology of China, 96 Jinzhao Rd., Hefei, Anhui, 230026, China. (qingling@mail.ustc.edu.cn).

respect to positive definite matrix  $\mathbf{A}$  is  $\|\mathbf{x}\|_{\mathbf{A}} := (\mathbf{x}^T \mathbf{A} \mathbf{x})^{1/2}$ . Given a function  $f$  its gradient evaluated at  $\mathbf{x}$  is denoted as  $\nabla f(\mathbf{x})$  and its Hessian as  $\nabla^2 f(\mathbf{x})$ . The diagonalized version of matrix  $\mathbf{A}$  is denoted by  $\text{diag}(\mathbf{A})$  where its diagonal components are identical with those of  $\mathbf{A}$  and the other components are null.

## II. PROBLEM FORMULATION

Consider  $\mathbf{x}_i \in \mathbb{R}^p$  as the copy of the decision variable  $\tilde{\mathbf{x}}$  at node  $i$  and define  $\mathcal{N}_i$  as the neighborhood of node  $i$ . Connectivity of the network implies that problem (1) is equivalent to the optimization problem

$$\begin{aligned} \{\mathbf{x}_{i,t}^*\}_{i=1}^n &:= \underset{\{\mathbf{x}_i\}_{i=1}^n}{\text{argmin}} \sum_{i=1}^n f_{i,t}(\mathbf{x}_i), \\ \text{s.t. } \mathbf{x}_i &= \mathbf{x}_j, \quad \text{for all } i, j \in \mathcal{N}_i. \end{aligned} \quad (2)$$

To verify the equivalence of (1) and (2), note that a set of feasible solutions for (2) has the general form of  $\mathbf{x}_1 = \dots = \mathbf{x}_n$ , since the network is connected. Likewise, the optimal solution of (2) satisfies  $\mathbf{x}_{1,t}^* = \dots = \mathbf{x}_{n,t}^*$ . When the arguments  $\mathbf{x}_i$  of the functions  $f_{i,t}(\mathbf{x}_i)$  are equal to each other the objective function  $\sum_{i=1}^n f_{i,t}(\mathbf{x}_i)$  in (2) can be simplified as the aggregate function  $\sum_{i=1}^n f_{i,t}(\mathbf{x})$  in (1). Thus, the optimal argument of each node  $\mathbf{x}_{i,t}^*$  in (2) is identical to the optimal solution  $\tilde{\mathbf{x}}_t^*$  of (1), i.e.,  $\mathbf{x}_{1,t}^* = \dots = \mathbf{x}_{n,t}^* = \tilde{\mathbf{x}}_t^*$ .

To derive the update for the dynamic ESOM algorithm, define  $\mathbf{x} := [\mathbf{x}_1; \dots; \mathbf{x}_n] \in \mathbb{R}^{np}$  as the concatenation of the local decision variables  $\mathbf{x}_i$  and the global function  $f_t : \mathbb{R}^{np} \rightarrow \mathbb{R}$  at time  $t$  as  $f_t(\mathbf{x}) = f_t(\mathbf{x}_1, \dots, \mathbf{x}_n) := \sum_{i=1}^n f_{i,t}(\mathbf{x}_i)$ . Further, we introduce the weight matrix  $\mathbf{W} \in \mathbb{R}^{n \times n}$  where the element  $w_{ij} \geq 0$  represents the weight that node  $i$  assigns to node  $j$ . The weight  $w_{ij}$  is nonzero if and only if  $j \in \mathcal{N}_i$  or  $j = i$ . We assume that the assigned weights are chosen such that the weight matrix  $\mathbf{W}$  satisfies the following conditions

$$\mathbf{W} = \mathbf{W}^T, \quad \mathbf{W}\mathbf{1} = \mathbf{1}, \quad \text{null}(\mathbf{I} - \mathbf{W}) = \text{span}(\mathbf{1}). \quad (3)$$

The first condition  $\mathbf{W} = \mathbf{W}^T$  implies that the weights are symmetric, i.e.,  $w_{ij} = w_{ji}$ . The condition  $\mathbf{W}\mathbf{1} = \mathbf{1}$  ensures that the weight matrix  $\mathbf{W}$  is doubly stochastic and the matrix  $\mathbf{I} - \mathbf{W}$  has a zero eigenvalue where its corresponding eigenvector is vector  $\mathbf{1}$ . The last condition  $\text{null}(\mathbf{I} - \mathbf{W}) = \text{span}(\mathbf{1})$  ensures that the matrix  $\mathbf{I} - \mathbf{W}$  has rank  $n-1$  and the condition  $(\mathbf{I} - \mathbf{W})\mathbf{v} = \mathbf{0}$  holds if and only if  $\mathbf{v} \in \text{span}\{\mathbf{1}\}$ . Conditions in (3) are typical of mixing matrices and they are required to enforce consensus.

It has been shown (Proposition 1 in [25]), if we define the matrix  $\mathbf{Z} = \mathbf{W} \otimes \mathbf{I}_p \in \mathbb{R}^{np \times np}$  as the Kronecker product of the weight matrix  $\mathbf{W}$  and the identity matrix  $\mathbf{I}_p$ , the optimization problem in (2) can be written as

$$\mathbf{x}_t^* = \underset{\mathbf{x} \in \mathbb{R}^{np}}{\text{argmin}} f_t(\mathbf{x}) \quad \text{s.t. } (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{x} = \mathbf{0}. \quad (4)$$

Thus, the optimization problem in (4) is equivalent to the original dynamic problem in (1) and we proceed to develop dynamic ESOM to solve (4) in lieu of (1). By introducing  $\mathbf{v} \in \mathbb{R}^{np}$  as the dual variable associated with the constraint

$(\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{x} = \mathbf{0}$  in (4), we define the augmented Lagrangian  $\mathcal{L}_t(\mathbf{x}, \mathbf{v})$  of (4) as

$$\mathcal{L}_t(\mathbf{x}, \mathbf{v}) = f_t(\mathbf{x}) + \mathbf{v}^T (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{x} + \frac{\alpha}{2} \mathbf{x}^T (\mathbf{I} - \mathbf{Z}) \mathbf{x}, \quad (5)$$

where  $\alpha$  is a positive constant. Based on the properties of the matrix  $\mathbf{Z}$ , the inner product  $\mathbf{x}^T (\mathbf{I} - \mathbf{Z}) \mathbf{x}$  augmented to the Lagrangian is null when the variable  $\mathbf{x}$  is a feasible solution of (4), otherwise the inner product is positive and behaves as a penalty for the violation of the consensus constraint.

A well studied approach to estimate the instantaneous minimizer  $\mathbf{x}_t^*$  is to define  $\mathbf{x}_t$  as the minimizer of the proximal augmented Lagrangian which is the sum of the augmented Lagrangian  $\mathcal{L}_t(\mathbf{x}, \mathbf{v}_{t-1})$  and the proximal term  $(\epsilon/2) \|\mathbf{x} - \mathbf{x}_{t-1}\|^2$ . This scheme can be interpreted as a dynamic extension of the proximal method of multipliers [26], [27]. Thus, the estimator  $\mathbf{x}_t$  is the minimizer of the optimization problem

$$\mathbf{x}_t = \underset{\mathbf{x} \in \mathbb{R}^{np}}{\text{argmin}} \left\{ \mathcal{L}_t(\mathbf{x}, \mathbf{v}_{t-1}) + \frac{\epsilon}{2} \|\mathbf{x} - \mathbf{x}_{t-1}\|^2 \right\}, \quad (6)$$

where  $\mathbf{v}_{t-1}$  is the dual variable evaluated at step  $t-1$  and  $\epsilon$  is a positive constant. The updated dual variable  $\mathbf{v}_t$  is updated by ascending through the augmented Lagrangian gradient  $\nabla_{\mathbf{v}} \mathcal{L}_t(\mathbf{x}_t, \mathbf{v}_{t-1})$  with respect to  $\mathbf{v}$  with stepsize  $\alpha$ ,

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \alpha (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{x}_t. \quad (7)$$

However, there are two issues with the updates in (6). The first issue is the computation time of the update, since the minimization could be computationally costly. The second drawback is the quadratic term  $\mathbf{x}^T (\mathbf{I} - \mathbf{Z}) \mathbf{x}$  in (6) which is not separable. Thus, the update is not implementable in a decentralized fashion. To resolve these issues we introduce the dynamic ESOM algorithm in the following section.

## III. DYNAMIC ESOM

In this section, we introduce the dynamic ESOM algorithm as a decentralized algorithm that replaces the augmented Lagrangian  $\mathcal{L}_t(\mathbf{x}, \mathbf{v}_{t-1})$  in (6) by its quadratic approximation. This modification reduces the computational complexity of the update in (6) and leads to a separable primal update. In particular, we approximate the augmented Lagrangian  $\mathcal{L}_t(\mathbf{x}, \mathbf{v}_{t-1})$  in (6) by its second-order Taylor's expansion near the point  $(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})$  which is given by  $\mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) + \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})^T (\mathbf{x} - \mathbf{x}_{t-1}) + (1/2) (\mathbf{x} - \mathbf{x}_{t-1})^T \nabla_{\mathbf{xx}}^2 \mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) (\mathbf{x} - \mathbf{x}_{t-1})$ . Applying this substitution leads to the update

$$\begin{aligned} \mathbf{x}_t = & \underset{\mathbf{x} \in \mathbb{R}^{np}}{\text{argmin}} \left\{ \mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) + \nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})^T (\mathbf{x} - \mathbf{x}_{t-1}) \right. \\ & \left. + \frac{1}{2} (\mathbf{x} - \mathbf{x}_{t-1})^T (\nabla_{\mathbf{xx}}^2 \mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1}) + \epsilon \mathbf{I}) (\mathbf{x} - \mathbf{x}_{t-1}) \right\}. \end{aligned} \quad (8)$$

Solving the minimization in the right hand side of (8) and using the definition of the augmented Lagrangian  $\mathcal{L}_t(\mathbf{x}, \mathbf{v})$

in (5), it follows that the variable  $\mathbf{x}_t$  can be evaluated as

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \mathbf{H}_t^{-1} \left[ \nabla f_t(\mathbf{x}_{t-1}) + (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{v}_{t-1} + \alpha(\mathbf{I} - \mathbf{Z})\mathbf{x}_{t-1} \right], \quad (9)$$

where the matrix  $\mathbf{H}_t \in \mathbb{R}^{np \times np}$  is defined as the Hessian of the objective function in (8) which is given by

$$\mathbf{H}_t := \nabla^2 f_t(\mathbf{x}_{t-1}) + \alpha(\mathbf{I} - \mathbf{Z}) + \epsilon \mathbf{I}. \quad (10)$$

The Hessian  $\mathbf{H}_t$  in (10) is a block neighbor sparse matrix. In other words, its  $(i, j)$ th block, which is in  $\mathbb{R}^{p \times p}$ , is non-zero if and only if  $j \in \mathcal{N}_i$  or  $j = i$ . This is true since the matrix  $\nabla^2 f_t(\mathbf{x}_{t-1}) + \epsilon \mathbf{I}$  is block diagonal and the matrix  $\alpha(\mathbf{I} - \mathbf{Z})$  is block neighbor sparse. Albeit, the Hessian  $\mathbf{H}_t$  is block neighbor sparse, its inverse  $\mathbf{H}_t^{-1}$  in (9) is not. Thus, the nodes cannot implement the update in (9) in a decentralized fashion.

To resolve this issue, we use a Hessian inverse approximation that is built on truncating the Taylor's series of the Hessian inverse  $\mathbf{H}_t^{-1}$  as in [17]. To be precise, we decompose the Hessian as  $\mathbf{H}_t = \mathbf{D}_t - \mathbf{B}$  where  $\mathbf{D}_t$  is a block diagonal positive definite matrix and  $\mathbf{B}$  is a neighbor sparse positive semidefinite matrix. We define the matrix  $\mathbf{D}_t$  as

$$\mathbf{D}_t := \nabla^2 f_t(\mathbf{x}_{t-1}) + \epsilon \mathbf{I} + 2\alpha(\mathbf{I} - \mathbf{Z}_d), \quad (11)$$

where  $\mathbf{Z}_d := \text{diag}(\mathbf{Z})$ . Hence, the relation  $\mathbf{B} = \mathbf{D}_t - \mathbf{H}_t$  implies that

$$\mathbf{B} := \alpha(\mathbf{I} - 2\mathbf{Z}_d + \mathbf{Z}). \quad (12)$$

Considering the decomposition  $\mathbf{H}_t = \mathbf{D}_t - \mathbf{B}$ , it follows that the Hessian inverse  $\mathbf{H}_t^{-1} = (\mathbf{D}_t - \mathbf{B})^{-1}$  can be written as  $\mathbf{H}_t^{-1} = \mathbf{D}_t^{-1/2}(\mathbf{I} - \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2})^{-1}\mathbf{D}_t^{-1/2}$  by factoring  $\mathbf{D}_t^{1/2}$  from both sides. Note that the absolute value of the eigenvalues of the matrix  $\mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$  are strictly smaller than 1; see e.g. Proposition 2 in [17]. Thus, we can use the Taylor's series  $(\mathbf{I} - \mathbf{X})^{-1} = \sum_{u=0}^{\infty} \mathbf{X}^u$  for  $\mathbf{X} = \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2}$  to write the Hessian inverse  $\mathbf{H}_t^{-1}$  as

$$\mathbf{H}_t^{-1} := \mathbf{D}_t^{-1/2} \sum_{u=0}^{\infty} \left( \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2} \right)^u \mathbf{D}_t^{-1/2}. \quad (13)$$

Computation of the Hessian inverse  $\mathbf{H}_t^{-1}$  in (13) requires infinite rounds of communication between the nodes; however, we can approximate the Hessian inverse  $\mathbf{H}_t^{-1}$  by truncating the first  $K+1$  terms of the sum in (13). This approximation leads to the Hessian inverse approximation

$$\hat{\mathbf{H}}_t^{-1}(K) := \mathbf{D}_t^{-1/2} \sum_{u=0}^K \left( \mathbf{D}_t^{-1/2}\mathbf{B}\mathbf{D}_t^{-1/2} \right)^u \mathbf{D}_t^{-1/2}. \quad (14)$$

The approximate Hessian inverse  $\hat{\mathbf{H}}_t^{-1}(K)$  is  $K$ -hop block neighbor sparse, i.e., its  $(i, j)$ th block is nonzero if and only if there exists at least one path between nodes  $i$  and  $j$  of length  $K$  or smaller.

We introduce the dynamic ESOM algorithm as a second-order method for solving the decentralized consensus opti-

mization problem which substitutes the Hessian inverse  $\mathbf{H}_t^{-1}$  in (9) by the  $K$ -hop block neighbor sparse Hessian inverse approximation  $\hat{\mathbf{H}}_t^{-1}(K)$  defined in (14). Thus, the update for the primal variable of dynamic ESOM is given by

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \hat{\mathbf{H}}_t^{-1}(K) \left[ \nabla f_t(\mathbf{x}_{t-1}) + (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{v}_{t-1} + \alpha(\mathbf{I} - \mathbf{Z})\mathbf{x}_{t-1} \right]. \quad (15)$$

The update for the dual variable  $\mathbf{v}_t$  of dynamic ESOM is identical to the update in (7),

$$\mathbf{v}_t = \mathbf{v}_{t-1} + \alpha(\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{x}_t. \quad (16)$$

Note that the primal and dual updates of dynamic ESOM in (15) and (16) are different from the updates of ESOM in [25] which is designed for static consensus optimization. In particular, the primal and dual updates of ESOM are derived by approximating the time-invariant augmented Lagrangian  $\mathcal{L}(\mathbf{x}, \mathbf{v}) := f(\mathbf{x}) + \mathbf{v}^T(\mathbf{I} - \mathbf{Z})^{1/2}\mathbf{x} + (\alpha/2)\mathbf{x}^T(\mathbf{I} - \mathbf{Z})\mathbf{x}$ , while the updates for dynamic ESOM are established by a quadratic approximation of the time-variant augmented Lagrangian  $\mathcal{L}_t(\mathbf{x}, \mathbf{v})$  defined in (5).

The updates in (15) and (16) explain the rationale behind dynamic ESOM; however, they are not implementable in a decentralized fashion, since the squared matrix  $(\mathbf{I} - \mathbf{Z})^{1/2}$  is not block neighbor sparse. In the following section, we introduce a new set of updates for dynamic ESOM which are implementable in a distributed fashion, while they are equivalent to the updates in (15) and (16).

#### A. Decentralized implementation of dynamic ESOM

To come up with updates for dynamic ESOM that can be implemented in a decentralized setting, define the sequence of variables  $\mathbf{q}_t$  as  $\mathbf{q}_t := (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{v}_t$ . Substitute the term  $(\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{v}_t$  in (15) by  $\mathbf{q}_t$  to rewrite the primal update as

$$\mathbf{x}_t = \mathbf{x}_{t-1} - \hat{\mathbf{H}}_t^{-1}(K) [\nabla f_t(\mathbf{x}_{t-1}) + \mathbf{q}_{t-1} + \alpha(\mathbf{I} - \mathbf{Z})\mathbf{x}_{t-1}]. \quad (17)$$

Multiplying the dual update in (16) by  $(\mathbf{I} - \mathbf{Z})^{1/2}$  from the left hand side and using the definition  $\mathbf{q}_t := (\mathbf{I} - \mathbf{Z})^{1/2} \mathbf{v}_t$ , it follows that

$$\mathbf{q}_t = \mathbf{q}_{t-1} + \alpha(\mathbf{I} - \mathbf{Z})\mathbf{x}_t. \quad (18)$$

The system of updates in (17) and (18) are implementable in a decentralized fashion, since the matrix  $\mathbf{I} - \mathbf{Z}$ , which is required for both updates, is block neighbors sparse. Notice that the updates in (17) and (18) are equivalent to the updates in (15) and (16), i.e., the sequence of iterates  $\mathbf{x}_t$  generated by these two schemes are identical.

We proceed to derive the local updates at each node to implement the primal and dual updates in (17) and (18), respectively. To do so, define  $\mathbf{g}_t$  as the augmented Lagrangian gradient  $\nabla_{\mathbf{x}} \mathcal{L}_t(\mathbf{x}_{t-1}, \mathbf{v}_{t-1})$  with respect to  $\mathbf{x}$  which is given by

$$\mathbf{g}_t = \nabla f_t(\mathbf{x}_{t-1}) + \mathbf{q}_{t-1} + \alpha(\mathbf{I} - \mathbf{Z})\mathbf{x}_{t-1} \quad (19)$$

Further, define the primal descent direction  $\mathbf{d}_t^{(K)}$  evaluated using the Hessian inverse approximation  $\hat{\mathbf{H}}_t^{-1}(K)$  with  $K$  levels of approximation as

$$\mathbf{d}_t^{(K)} := -\hat{\mathbf{H}}_t^{-1}(K)\mathbf{g}_t. \quad (20)$$

The definition of the descent direction  $\mathbf{d}_t^{(K)}$  in (20) allows us to rewrite the update in (17) as  $\mathbf{x}_t = \mathbf{x}_{t-1} + \mathbf{d}_t^{(K)}$ . According to the mechanism of Hessian inverse approximation in (14), the descent directions  $\mathbf{d}_t^{(k)}$  and  $\mathbf{d}_t^{(k+1)}$  satisfy the recursion

$$\mathbf{d}_t^{(k+1)} = \mathbf{D}_t^{-1}\mathbf{B}\mathbf{d}_t^{(k)} - \mathbf{D}_t^{-1}\mathbf{g}_t. \quad (21)$$

Consider  $\mathbf{d}_{i,t-1}^{(k)}$  as the descent direction of node  $i$  at step  $t$  which is the  $i$ -th element of the global descent direction  $\mathbf{d}_t^{(k)} = [\mathbf{d}_{1,t}^{(k)}; \dots; \mathbf{d}_{n,t}^{(k)}]$ . Use this definition to write the localized version of the relation in (21) at node  $i$  as

$$\mathbf{d}_{i,t}^{(k+1)} = \mathbf{D}_{ii,t}^{-1} \sum_{j=i,j \in \mathcal{N}_i} (\mathbf{B}_{ij}\mathbf{d}_{j,t}^{(k)}) - \mathbf{D}_{ii,t}^{-1}\mathbf{g}_{i,t}, \quad (22)$$

where  $\mathbf{D}_{ii,t}$  is the  $i$ -th diagonal block of the matrix  $\mathbf{D}_t$  and  $\mathbf{B}_{ij}$  is the  $(i,j)$ -th block of the matrix  $\mathbf{B}$ . Based on the expression in (22), node  $i$  is able to compute its descent direction  $\mathbf{d}_{i,t}^{(k+1)}$  using the  $k$ -th level descent direction of itself  $\mathbf{d}_{i,t}^{(k)}$  and its neighbors  $\mathbf{d}_{j,t}^{(k)}$  for  $j \in \mathcal{N}_i$ . Therefore, if nodes exchange their  $k$ -th level descent direction  $\mathbf{d}_{i,t}^{(k)}$  with their neighbors, they can compute the  $(k+1)$ -th level descent direction  $\mathbf{d}_{i,t}^{(k+1)}$ .

Notice that the block  $\mathbf{D}_{ii,t} := \nabla^2 f_{i,t}(\mathbf{x}_{i,t-1}) + (2\alpha(1 - w_{ii}))\mathbf{I} + \epsilon\mathbf{I}$  is locally available at node  $i$ . Moreover, node  $i$  can evaluate the blocks  $\mathbf{B}_{ii} = \alpha(1 - w_{ii})\mathbf{I}$  and  $\mathbf{B}_{ij} = \alpha w_{ij}\mathbf{I}$  locally. In addition, nodes can compute the gradient  $\mathbf{g}_t$  by communicating with their neighbors. To confirm this claim, observe that the  $i$ -th element of the gradient  $\mathbf{g}_t = [\mathbf{g}_{1,t}; \dots; \mathbf{g}_{n,t}]$  associated with node  $i$  is given by

$$\mathbf{g}_{i,t} := \nabla f_t(\mathbf{x}_{i,t-1}) + \mathbf{q}_{i,t-1} + \alpha(1 - w_{ii})\mathbf{x}_{i,t-1} - \alpha \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{x}_{j,t-1}, \quad (23)$$

where  $\mathbf{q}_{i,t-1} \in \mathbb{R}^p$  is the  $i$ -th element of the vector  $\mathbf{q}_{t-1} = [\mathbf{q}_{1,t-1}; \dots; \mathbf{q}_{n,t-1}] \in \mathbb{R}^{np}$ . Hence, node  $i$  can compute its local gradient  $\mathbf{g}_{i,t}$  using local information  $\mathbf{q}_{i,t-1}$  and  $\mathbf{x}_{i,t-1}$ , and its neighbors' information  $\mathbf{x}_{j,t-1}$  where  $j \in \mathcal{N}_i$ .

The recursive update in (22) shows that at each step  $t$  nodes can compute the descent directions  $\mathbf{d}_{i,t}^{(K)}$  by  $K$  rounds of communication with their neighbors. Moreover, observe that nodes can implement the dual update in (18) as

$$\mathbf{q}_{i,t} = \mathbf{q}_{i,t-1} + \alpha(1 - w_{ii})\mathbf{x}_{i,t} - \alpha \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{x}_{j,t}, \quad (24)$$

by having access to the updated primal variables  $\mathbf{x}_{j,t}$  of their neighbors  $j \in \mathcal{N}_i$ .

The steps of dynamic ESOM- $K$  at node  $i$  are summarized in Algorithm 1. In step 3, each node  $i$  observes its local function  $f_{i,t}$  for the current time  $t$  and uses this information to compute the block  $\mathbf{D}_{ii,t}$  and the local gradient  $\mathbf{g}_{i,t}$  in Steps 4 and 5, respectively. Node  $i$  computes its  $(k+1)$ -

---

#### Algorithm 1 Dynamic ESOM- $K$ method at node $i$

---

**Require:** Initial iterates  $\mathbf{x}_{i,0} = \mathbf{x}_{j,0} = \mathbf{0}$  for  $j \in \mathcal{N}_i$  and  $\mathbf{q}_{i,0} = \mathbf{0}$ .  
1: Compute  $\mathbf{B}_{ii} = \alpha(1 - w_{ii})\mathbf{I}_p$  and  $\mathbf{B}_{ij} = \alpha w_{ij}\mathbf{I}_p$  all  $j \in \mathcal{N}_i$   
2: **for** times  $t = 1, 2, \dots$  **do**  
3:   Observe the local function  $f_{i,t}$   
4:   Compute  $\mathbf{D}_{ii,t} = \nabla^2 f_{i,t}(\mathbf{x}_{i,t-1}) + 2\alpha(1 - w_{ii})\mathbf{I}_p$   
5:   Compute the gradient  $\mathbf{g}_{i,t}$  as in (23)  
6:   Compute the initial descent direction  $\mathbf{d}_{i,t}^{(0)} = -\mathbf{D}_{ii,t}^{-1}\mathbf{g}_{i,t}$   
7:   **for**  $k = 0, \dots, K-1$  **do**  
8:     Exchange  $\mathbf{d}_{i,t-1}^{(k)}$  with neighbors  $j \in \mathcal{N}_i$   
9:     Compute  $\mathbf{d}_{i,t}^{(k+1)} = \mathbf{D}_{ii,t}^{-1} \sum_{j=i,j \in \mathcal{N}_i} \mathbf{B}_{ij}\mathbf{d}_{j,t}^{(k)} - \mathbf{D}_{ii,t}^{-1}\mathbf{g}_{i,t}$   
10:   **end for**  
11:   Update primal iterate:  $\mathbf{x}_{i,t} = \mathbf{x}_{i,t-1} + \mathbf{d}_{i,t}^{(K)}$   
12:   Exchange iterates  $\mathbf{x}_{i,t}$  with neighbors  $j \in \mathcal{N}_i$ .  
13:   Update the dual iterate:  
     $\mathbf{q}_{i,t} = \mathbf{q}_{i,t-1} + \alpha(1 - w_{ii})\mathbf{x}_{i,t} - \alpha \sum_{j \in \mathcal{N}_i} w_{ij}\mathbf{x}_{j,t}$   
14: **end for**

---

th level descent direction  $\mathbf{d}_{i,t}^{(k+1)}$  in Step 9 using the  $k$ -th level local descent direction  $\mathbf{d}_{i,t}^{(k)}$  and the neighbors' decent directions  $\mathbf{d}_{j,t}^{(k)}$  which are exchanged in Step 8. Note that the recursion in Steps 8 and 9 are initialized by the descent direction  $\mathbf{d}_{i,t}^{(0)}$  of dynamic ESOM-0 evaluated in Step 6. Each node computes its local primal variable  $\mathbf{x}_{i,t}$  in Step 11 and exchanges it with its neighbors in Step 12. The dual variables  $\mathbf{q}_{i,t}$  can be updated in Step 13, using the updated local and neighboring primal variables. The blocks  $\mathbf{B}_{ij}$  for  $j \in \mathcal{N}_i$  and  $j = i$  are time invariant and they are computed and stored locally in Step 1.

**Remark 1** One may raise the question about the choice of  $K$  for dynamic ESOM- $K$ . Note that the implementation of dynamic ESOM- $K$  requires  $K+1$  rounds of communication between neighboring nodes. Thus, by increasing the choice of  $K$  the computation time of the algorithm increases. Although, larger choice of  $K$  leads to a better Hessian inverse approximation and faster convergence, the required time may exceed the time between the subsequent instances  $t-1$  and  $t$ . Therefore, based on the available time between the consecutive times  $t-1$  and  $t$ , we should pick the largest choice of  $K$  which is affordable in terms of computation and communication time.

#### IV. CONVERGENCE ANALYSIS

In this section we study the difference between the sequence of the iterates  $\mathbf{x}_t$  generated by dynamic ESOM and the sequence of the optimal arguments  $\mathbf{x}_t^* = [\mathbf{x}_{1,t}^*; \dots; \mathbf{x}_{n,t}^*] = [\tilde{\mathbf{x}}_t^*; \dots; \tilde{\mathbf{x}}_t^*]$ . To prove the results, we assume the following conditions are satisfied.

**Assumption 1** The instantaneous local objective functions  $f_{i,t}(\mathbf{x})$  are twice differentiable and the eigenvalues of the instantaneous local objective functions Hessian  $\nabla^2 f_{i,t}$  are bounded by positive constants  $0 < m \leq M < \infty$ , i.e.

$$m\mathbf{I} \preceq \nabla^2 f_{i,t}(\mathbf{x}_i) \preceq M\mathbf{I}, \quad (25)$$

for all  $\mathbf{x}_i \in \mathbb{R}^p$  and  $i = 1, \dots, n$ .

**Assumption 2** The instantaneous local objective functions Hessian  $\nabla^2 f_{i,t}$  are Lipschitz continuous with constant  $L$ ,

$$\|\nabla^2 f_{i,t}(\mathbf{x}_i) - \nabla^2 f_{i,t}(\mathbf{y}_i)\| \leq L\|\mathbf{x}_i - \mathbf{y}_i\|, \quad (26)$$

for all  $\mathbf{x}_i, \mathbf{y}_i \in \mathbb{R}^p$  and  $i = 1, \dots, n$ .

We can interpret the lower and upper bounds on the eigenvalues of the Hessians  $\nabla^2 f_{i,t}$  as the strong convexity of the instantaneous local functions  $f_{i,t}$  with constant  $m$  and the Lipschitz continuity of the instantaneous local gradients  $\nabla f_{i,t}$  with constant  $M$ , respectively. The global objective function Hessian  $\nabla^2 f_t(\mathbf{x})$  at step  $t$  is a block diagonal matrix where its  $i$ -th diagonal block is  $\nabla^2 f_{i,t}(\mathbf{x}_i)$ . Hence, the bounds in (25) for the eigenvalues of the instantaneous local Hessians also hold for the instantaneous global Hessian  $\nabla^2 f_t(\mathbf{x})$ , i.e.,

$$m\mathbf{I} \preceq \nabla^2 f_t(\mathbf{x}) \preceq M\mathbf{I}, \quad (27)$$

for all  $\mathbf{x} \in \mathbb{R}^{np}$ . Thus, the global objective function  $f_t$  is also strongly convex with constant  $m$  and its gradients  $\nabla f_t$  are Lipschitz continuous with constant  $M$ . Likewise, the Lipschitz continuity of the local Hessians  $\nabla^2 f_{i,t}$ , which is a customary assumption in the analysis of second-order methods, implies that the instantaneous global Hessian  $\nabla^2 f_t$  is also Lipschitz continuous with constant  $L$ , i.e.,

$$\|\nabla^2 f_t(\mathbf{x}) - \nabla^2 f_t(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \quad (28)$$

for any  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{np}$  – see e.g., Lemma 1 in [17].

To characterize the error of dynamic ESOM, we define the vector  $\mathbf{u}_t = [\mathbf{x}_t; \mathbf{v}_t] \in \mathbb{R}^{2np}$  as the concatenation of the primal and dual iterates at step  $t$ . Likewise, we define  $\mathbf{u}_t^* = [\mathbf{x}_t^*; \mathbf{v}_t^*] \in \mathbb{R}^{2np}$  as the concatenation of the optimal arguments at time  $t$ . We proceed to characterize an upper bound for the error sequence  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$  where the positive definite matrix  $\mathbf{G}$  is defined as  $\mathbf{G} := \text{diag}(\mathbf{I}_{np}, \epsilon\alpha\mathbf{I}_{np}) \in \mathbb{R}^{2np \times 2np}$ . In the following lemma we establish an upper bound for the norm  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$  in terms of the difference between the previous vector  $\mathbf{u}_{t-1}$  and the current optimal argument  $\mathbf{u}_t^*$ .

**Lemma 1** Consider the updates of dynamic ESOM as introduced in (15)-(16) and recall the definitions of the vector  $\mathbf{u}$  and matrix  $\mathbf{G}$ . If Assumptions 1 and 2 hold, then there exists a positive scalar  $0 < \delta$  such that the sequence of iterates  $\mathbf{u}_t$  generated by dynamic ESOM satisfies

$$\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{1}{\sqrt{1+\delta}} \|\mathbf{u}_{t-1} - \mathbf{u}_t^*\|_{\mathbf{G}}. \quad (29)$$

**Proof:** The proof can be established by following the steps of the proof of Theorem 2 in [25]. ■

The constant  $\delta$  in (29) is a function of the objective function  $f_t$  parameters, network topology, and level of Hessian inverse approximation  $K$ . In particular, the constant  $\delta$  is close to zero when the objective function is ill-conditioned, or the

network is not well connected. Moreover, larger choice of  $K$  leads to a larger choice of  $\delta$  which leads to a smaller error  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$ .

The result in Lemma 1 illustrates that the iterate  $\mathbf{u}_t$  is closer to the optimal argument  $\mathbf{u}_t^*$  at step  $t$  relative to the previous iterate  $\mathbf{u}_{t-1}$ . This result is implied from the fact that  $\mathbf{u}_t$  is evaluated based on the observed function  $f_t$  at step  $t$ . Based on the result in Lemma 1, we can establish an upper bound for the error  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$  at step  $t$  in terms of the error of the previous time  $\|\mathbf{u}_{t-1} - \mathbf{u}_{t-1}^*\|_{\mathbf{G}}$  and the variation of the optimal arguments. We characterize this upper bound in the following theorem.

**Theorem 1** Consider the dynamic ESOM algorithm as introduced in (15)-(16) and recall the definitions of the vector  $\mathbf{u}$  and matrix  $\mathbf{G}$ . Define  $\gamma$  as the smallest non-zero eigenvalue of the positive semidefinite matrix  $\mathbf{I} - \mathbf{Z}$ . Further, define the dynamic optimality drift  $d_t$  as

$$d_t := \|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\| + \frac{\sqrt{\alpha\epsilon}}{\sqrt{\gamma}} \|\nabla f_t(\mathbf{x}_t^*) - \nabla f_{t-1}(\mathbf{x}_{t-1}^*)\|. \quad (30)$$

If Assumptions 1 and 2 hold, then the sequence of iterates  $\mathbf{u}_t$  generated by dynamic ESOM satisfies

$$\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{1}{\sqrt{1+\delta}} \|\mathbf{u}_{t-1} - \mathbf{u}_{t-1}^*\|_{\mathbf{G}} + \frac{d_t}{\sqrt{1+\delta}}. \quad (31)$$

**Proof:** See Appendix VII-A. ■

The optimality drift  $d_t$  captures the drift between the two consecutive optimal arguments  $\mathbf{x}_t^*$  and  $\mathbf{x}_{t-1}^*$  as well as the difference between the two successive optimal gradients  $\nabla f_t(\mathbf{x}_t^*)$  and  $\nabla f_{t-1}(\mathbf{x}_{t-1}^*)$ . The result in Theorem 1 shows that the sequence of the error  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$  approaches linearly a steady state error bound. Note that the optimality drift  $d_t$  is small when the functions  $f_t$  change sufficiently slow. The result in (31) is consistent with the results for the static version of the optimization problem in (1). In the static setting, where  $\mathbf{u}_t^* = \mathbf{u}_{t-1}^* = \mathbf{u}^*$ ,  $\mathbf{x}_t^* = \mathbf{x}_{t-1}^* = \mathbf{x}^*$ , and  $\nabla f_t(\mathbf{x}_t^*) = \nabla f_{t-1}(\mathbf{x}_{t-1}^*) = \nabla f(\mathbf{x}^*)$ , the result in (31) can be simplified as  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq (1/\sqrt{1+\delta})\|\mathbf{u}_{t-1} - \mathbf{u}_t^*\|_{\mathbf{G}}$  which shows linear convergence of the iterates generated by ESOM to the optimal argument.

In the following theorem we use the result in Theorem 1 to show that the error of dynamic ESOM, which is characterized by the norm  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$ , approaches a steady state error.

**Theorem 2** Consider the dynamic ESOM algorithm as introduced in (15)-(16) and recall the definition of the optimality drift  $d_t$  in (30). Further, define  $d_{\max} := \max_t d_t$  as the maximum of the optimality drift  $d_t$  for all times  $t$ . If Assumptions 1 and 2 hold, then the limit supremum of the sequence  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$  is bounded above by

$$\limsup_{t \rightarrow \infty} \|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{d_{\max}}{\sqrt{1+\delta} - 1}. \quad (32)$$

**Proof:** See Appendix VII-B. ■

The steady state error of the sequence generated by dynamic ESOM is characterized in Theorem 2. As we expect, if

the maximum optimality drift  $d_{\max}$  is not large the dynamic ESOM algorithm approaches a reasonable asymptotic error. Moreover, the steady state error is smaller for the case that the constant of linear convergence  $\delta$  is larger. This observation shows that the steady state error of ESOM- $K$  reduces by increasing the level of Hessian inverse approximation  $K$ . This is true, since for larger choice of  $K$ , the constant  $\delta$  is larger.

Convergence of the sequence  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$ , which characterizes the primal and dual errors of the iterates of dynamic ESOM, follows that the sequence of primal iterates  $\mathbf{x}_t$  converges to a neighborhood of the optimal argument  $\mathbf{x}_t^*$ . This is shown in the following corollary.

**Corollary 1** *Recall the definition of the maximum optimality drift  $d_{\max}$  and suppose that the conditions in Theorem 2 are satisfied. Then the primal error  $\|\mathbf{x}_t - \mathbf{x}_t^*\|$  of dynamic ESOM is upper bounded as*

$$\limsup_{t \rightarrow \infty} \|\mathbf{x}_t - \mathbf{x}_t^*\| \leq \frac{d_{\max}}{\sqrt{1 + \delta} - 1}. \quad (33)$$

**Proof :** Based on the definition of the norm  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$ , we can simplify the norm as  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} = [\|\mathbf{x}_t - \mathbf{x}_t^*\|^2 + \alpha\epsilon\|\mathbf{v}_t - \mathbf{v}_t^*\|^2]^{1/2}$ . According to this definition, we obtain that the primal error  $\|\mathbf{x}_t - \mathbf{x}_t^*\|$  is smaller than the norm  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$ . This observation in conjunction with the result in (32) implies the claim in (33). ■

## V. NUMERICAL EXPERIMENTS

In this section, we study the performance of the proposed dynamic ESOM method in solving a dynamic least squares problem. We consider a connected network with  $n = 20$  and connectivity ratio  $r_c = 0.15$ , i.e., edges are generated randomly with probability 0.15.

We consider a decentralized dynamic least squares problem where at time  $t$  nodes aim to estimate the true signal  $\tilde{\mathbf{x}}_t^* \in \mathbb{R}^5$ . Consider the linear model  $\mathbf{y}_{i,t} = \mathbf{H}_{i,t}\tilde{\mathbf{x}}_t^* + \boldsymbol{\eta}_{i,t}$  where the matrix  $\mathbf{H}_{i,t} \in \mathbb{R}^{5 \times 5}$  is a regressor matrix and the vector  $\boldsymbol{\eta}_{i,t} \in \mathbb{R}^5$  is an additive noise. We assume that node  $i$  observes the vector  $\mathbf{y}_{i,t}$  and collaborates with its neighbors to find the true signal  $\tilde{\mathbf{x}}_t^*$ . In other words, the nodes' goal is to solve the least squares problem

$$\tilde{\mathbf{x}}_t^* = \underset{\mathbf{x} \in \mathbb{R}^5}{\operatorname{argmin}} \sum_{i=1}^n \frac{1}{2} \|\mathbf{H}_{i,t}\mathbf{x} - \mathbf{y}_{i,t}\|^2. \quad (34)$$

Considering the definition of the global optimization problem in (34), the local objective function of node  $i$  at time  $t$  is given by  $f_{i,t} := (1/2)\|\mathbf{H}_{i,t}\mathbf{x} - \mathbf{y}_{i,t}\|^2$ .

We compare the dynamic variations of ESOM-0, ESOM-2, Network Newton-0 (NN-0) [17], and EXTRA [28] in solving the dynamic least squares problem in (34). In our experiments, we assume that the matrices  $\mathbf{H}_{i,t}$  are fixed over time, i.e.,  $\mathbf{H}_{i,t} = \mathbf{H}_i$ . We generate the components of  $\mathbf{H}_i$  following the Gaussian distribution  $\mathcal{N}(0, 1)$ . Although, the matrices  $\mathbf{H}_i$  are time-invariant, we assume that the vectors  $\mathbf{y}_{i,t}$  are changing over time. We assume that after every

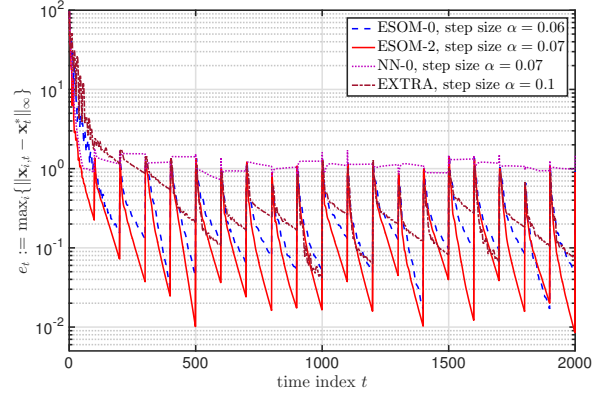


Fig. 1: The convergence path of  $e_t$  versus time index  $t$  for dynamic ESOM-0, ESOM-2, EXTRA, and NN-0. Dynamic ESOM-2 has the best performance among all the considered methods.

100 iterations the components of the vectors  $\mathbf{y}_{i,t}$  change in a way that the new global minimizer  $\tilde{\mathbf{x}}_t^*$  satisfies  $\tilde{\mathbf{x}}_t^* = |\sin(\pi t/500)|\tilde{\mathbf{x}}_0^*$ . In other words,  $\tilde{\mathbf{x}}_t^* = |\sin(\pi t/500)|\tilde{\mathbf{x}}_0^*$  if  $t$  is a multiplicant of 100, otherwise  $\tilde{\mathbf{x}}_t^* = \tilde{\mathbf{x}}_{t-1}^*$ . Moreover, we assume that every agent starts from the initial point  $\mathbf{x}_{i,0}$  that satisfies the condition  $\|\mathbf{x}_{i,0} - \tilde{\mathbf{x}}_0^*\| = 100$ .

We characterize error as the maximum difference between the coordinates of each node's variable and the optimal argument  $\tilde{\mathbf{x}}_t^*$ . Thus, if we define  $\mathbf{x}_{i,t}[s]$  as the  $s$ -th coordinate of the variable  $\mathbf{x}_{i,t}$ , the error is defined as  $e_t := \max_i \{\max_s \{|\mathbf{x}_{i,t}[s] - \tilde{\mathbf{x}}_t^*[s]|\}\}$ . The error  $e_t$  also can be written as

$$e_t := \max_i \{\|\mathbf{x}_{i,t} - \tilde{\mathbf{x}}_t^*\|_{\infty}\}, \quad (35)$$

using the definition of the infinity norm  $\|\cdot\|_{\infty}$ .

Figure 1 shows the error  $e_t$  versus the time index  $t$  for the four algorithms of interest. As we observe, during the time that the optimal argument is fixed, NN-0 approaches a neighborhood of the optimal solution and its error  $e_t$  stays constant, while EXTRA, ESOM-0, and ESOM-2 converge linearly to the exact solution and their error  $e_t$  diminish. It is also worth mentioning that both ESOM-0 and ESOM-2 outperform EXTRA by incorporating second-order information, and ESOM-2 has the best performance among all the considered methods. If more rounds of communication is affordable between the subsequent instances  $t$  and  $t+1$ , then the performance of dynamic ESOM- $K$  can be improved by using larger values for  $K$ . Note that whenever the optimal argument  $\mathbf{x}_t^*$  changes, which happens every 100 iterations, all the algorithms readjust and correct their descent direction to track the new optimal argument.

To study the performance of dynamic NN-0, EXTRA, ESOM-0, and ESOM-2 in more details, we compare the values of the first coordinate  $\mathbf{x}_{1,t}[1]$  of node 1 generated by these methods with the first coordinate of the optimal argument  $\tilde{\mathbf{x}}_t^*[1]$ . This comparison is shown in Figure 2. As we observe in Figure 2, all the dynamic methods are unable to track the true path in the first 200 iterations. Dynamic ESOM-0 and ESOM-2 can track the optimal argument after

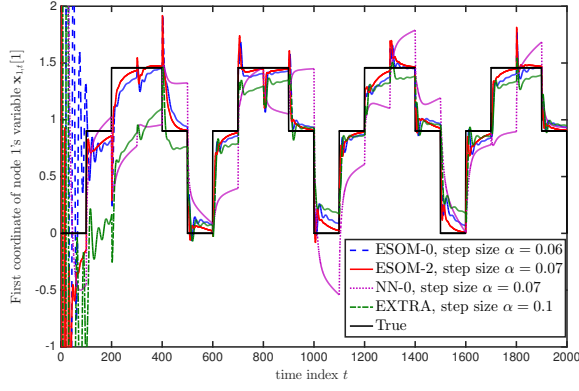


Fig. 2: Comparison of node 1's first coordinate for the iterates generated by dynamic variations of ESOM-0, ESOM-2, NN-0, and EXTRA with the first coordinate of the optimal argument  $\tilde{\mathbf{x}}_t^*$ .

the first 200 iterations, while the accuracy of dynamic ESOM-2 is higher relative to dynamic ESOM-0. Dynamic EXTRA starts tracking the true path after 400 iterations, while its error is worse than the ones for dynamic ESOM-0 and ESOM-2. For the dynamic NN-0 method, the error is always larger than the error of the other dynamic methods.

These observations verify the theoretical results in Section IV. To be more precise, they show that the dynamic variations of EXTRA and ESOM- $K$  outperform dynamic NN, since they converge linearly to the optimal arguments while NN converges to a neighborhood of the optimal solution in static settings. Moreover, dynamic ESOM- $K$ , irrespective to the choice of  $K$ , improves the performance of dynamic EXTRA by incorporating second-order information of the augmented Lagrangian in (5). Further, larger choice of  $K$  for dynamic ESOM- $K$  leads to a faster linear convergence, i.e., larger  $\delta$ , which implies a smaller steady state error. This observation verifies the result in Corollary 1.

## VI. CONCLUSIONS

We considered the application of the Exact Second-Order Methods (ESOM) in solving a dynamic consensus optimization problem where the local functions available at nodes are time-variant. The proposed dynamic ESOM method relies on the use of a separable quadratic approximation of a suitably defined time-varying augmented Lagrangian, and a truncated Taylor's series to estimate the solution of the first order condition imposed on the minimization of the quadratic approximation of the augmented Lagrangian. We proved that under proper assumptions, the sequence of iterates generated by dynamic ESOM converges linearly to a neighborhood of the sequence of optimal arguments. We characterized the steady state error in terms of the maximum difference between the successive optimal arguments  $\mathbf{x}_{t-1}^*$  and  $\mathbf{x}_t^*$  as well as the optimal gradients  $\nabla f_{t-1}(\mathbf{x}_{t-1}^*)$  and  $\nabla f_t(\mathbf{x}_t^*)$ . Numerical results showcase the advantages of the proposed dynamic ESOM method relative to existing dynamic decentralized methods.

## VII. APPENDIX

### A. Proof of Theorem 1

According to the definition of the vector  $\mathbf{u}$  and matrix  $\mathbf{G}$  we can write

$$\begin{aligned} \|\mathbf{u}_{t-1}^* - \mathbf{u}_t^*\|_{\mathbf{G}} &= [\|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\|^2 + \alpha\epsilon\|\mathbf{v}_{t-1}^* - \mathbf{v}_t^*\|^2]^{1/2} \\ &\leq \|\mathbf{x}_{t-1}^* - \mathbf{x}_t^*\| + \sqrt{\alpha\epsilon}\|\mathbf{v}_{t-1}^* - \mathbf{v}_t^*\|, \end{aligned} \quad (36)$$

where the inequality follows from the inequality  $a^2 + b^2 \leq (a + b)^2$  for positive scalars  $a$  and  $b$ . The KKT condition of the optimization problem in (4) yields

$$\nabla f_t(\mathbf{x}_t^*) + (\mathbf{I} - \mathbf{Z})^{1/2}\mathbf{v}_t^* = \mathbf{0}. \quad (37)$$

By writing the KKT condition in (37) for time  $t - 1$  we obtain that

$$\nabla f_t(\mathbf{x}_t^*) - \nabla f_{t-1}(\mathbf{x}_{t-1}^*) + (\mathbf{I} - \mathbf{Z})^{1/2}(\mathbf{v}_t^* - \mathbf{v}_{t-1}^*) = \mathbf{0}. \quad (38)$$

Since the vectors  $\mathbf{v}_t^*$  and  $\mathbf{v}_{t-1}^*$  are in the column space of the matrix  $\mathbf{I} - \mathbf{Z}$ , we obtain that  $(\mathbf{I} - \mathbf{Z})^{1/2}(\mathbf{v}_t^* - \mathbf{v}_{t-1}^*)$  is bounded below by  $\gamma^{1/2}\|\mathbf{v}_t^* - \mathbf{v}_{t-1}^*\|$ . This lower bound in conjunction with the expression in (38) implies that the norm  $\|\mathbf{v}_t^* - \mathbf{v}_{t-1}^*\|$  is bounded above as

$$\|\mathbf{v}_t^* - \mathbf{v}_{t-1}^*\| \leq \frac{1}{\sqrt{\gamma}}\|\nabla f_t(\mathbf{x}_t^*) - \nabla f_{t-1}(\mathbf{x}_{t-1}^*)\|. \quad (39)$$

By substituting the upper bound in (39) to (36), the inequality  $\|\mathbf{u}_{t-1}^* - \mathbf{u}_t^*\|_{\mathbf{G}} \leq d_t$  follows.

Based on the triangle inequality, the weighted norm  $\|\mathbf{u}_{t-1} - \mathbf{u}_t^*\|_{\mathbf{G}}$  is bounded above by the sum

$$\|\mathbf{u}_{t-1} - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \|\mathbf{u}_{t-1} - \mathbf{u}_{t-1}^*\|_{\mathbf{G}} + \|\mathbf{u}_{t-1}^* - \mathbf{u}_t^*\|_{\mathbf{G}}. \quad (40)$$

Since the norm  $\|\mathbf{u}_{t-1}^* - \mathbf{u}_t^*\|_{\mathbf{G}}$  is smaller than the drift  $d_t$  defined in (30), we can replace  $\|\mathbf{u}_{t-1}^* - \mathbf{u}_t^*\|_{\mathbf{G}}$  in (40) by  $d_t$

$$\|\mathbf{u}_{t-1} - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \|\mathbf{u}_{t-1} - \mathbf{u}_{t-1}^*\|_{\mathbf{G}} + d_t. \quad (41)$$

Combining the results in (41) and (29), the claim in (31) follows.

### B. Proof of Theorem 2

We prove the claim in (32) based on (31) in Theorem 1. First, consider the inequality in (31) for time  $t - 1$  which is given by

$$\|\mathbf{u}_{t-1} - \mathbf{u}_{t-1}^*\|_{\mathbf{G}} \leq \frac{1}{\sqrt{1+\delta}}\|\mathbf{u}_{t-2} - \mathbf{u}_{t-2}^*\|_{\mathbf{G}} + \frac{d_{t-1}}{\sqrt{1+\delta}}. \quad (42)$$

Substituting the norm  $\|\mathbf{u}_{t-1} - \mathbf{u}_{t-1}^*\|_{\mathbf{G}}$  in (31) by the upper bound in (42) yields

$$\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{\|\mathbf{u}_{t-2} - \mathbf{u}_{t-2}^*\|_{\mathbf{G}}}{(\sqrt{1+\delta})^2} + \frac{d_{t-1}}{(\sqrt{1+\delta})^2} + \frac{d_t}{\sqrt{1+\delta}}. \quad (43)$$

By considering the expression in (31) for all times  $s \leq t$  and recursively it follows that the error  $\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}}$  at step  $t$  and the initial error  $\|\mathbf{u}_0 - \mathbf{u}_0^*\|_{\mathbf{G}}$  satisfy

$$\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{\|\mathbf{u}_0 - \mathbf{u}_0^*\|_{\mathbf{G}}}{(\sqrt{1+\delta})^t} + \sum_{s=1}^t \frac{d_s}{(\sqrt{1+\delta})^{t-s+1}}. \quad (44)$$



Now considering the definition of  $d_{\max}$  as  $d_{\max} = \sup_{t \geq 1} d_t$ , we obtain that the sum in the right hand side of (44) is bounded above by

$$\begin{aligned} \sum_{s=1}^t \frac{d_s}{(\sqrt{1+\delta})^{t-s+1}} &\leq \sum_{s=1}^t \frac{d_{\max}}{(\sqrt{1+\delta})^{t-s+1}} \\ &\leq \frac{d_{\max}}{\sqrt{1+\delta}} \times \frac{1 - (\sqrt{1+\delta})^{-t}}{1 - (\sqrt{1+\delta})^{-1}}, \end{aligned} \quad (45)$$

where the second inequality is implied from the simplification  $\sum_{s=1}^t \rho^s = \rho(1 - \rho^t)/(1 - \rho)$  when  $\rho < 1$ . Replacing the sum in the right hand side of (44) by the upper bound in (45) yields

$$\|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{\|\mathbf{u}_0 - \mathbf{u}_0^*\|_{\mathbf{G}}}{(\sqrt{1+\delta})^t} + \frac{d_{\max}}{\sqrt{1+\delta}} \times \frac{1 - (\sqrt{1+\delta})^{-t}}{1 - (\sqrt{1+\delta})^{-1}}. \quad (46)$$

Taking  $t \rightarrow \infty$ , the term  $\|\mathbf{u}_0 - \mathbf{u}_0^*\|_{\mathbf{G}}/(\sqrt{1+\delta})^t$  in the right hand side of (46) vanishes. Moreover, the term  $(\sqrt{1+\delta})^{-t}$  approaches 0. From these observations it follows that

$$\limsup_{t \rightarrow \infty} \|\mathbf{u}_t - \mathbf{u}_t^*\|_{\mathbf{G}} \leq \frac{d_{\max}}{\sqrt{1+\delta} - 1}, \quad (47)$$

and the proof is complete.

## REFERENCES

- [1] P. Aliksson and A. Rantzer, "Distributed kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, 2006, pp. 2445–2450.
- [2] M. Farina, G. Ferrari-Trecate, R. Scattolini *et al.*, "Distributed moving horizon estimation for linear constrained systems," *IEEE Transactions on Automatic Control*, vol. 55, no. 11, pp. 2462–2475, 2010.
- [3] F. Y. Jakubiec and A. Ribeiro, "D-map: Distributed maximum a posteriori probability estimation of dynamic systems," *Signal Processing, IEEE Transactions on*, vol. 61, no. 2, pp. 450–466, 2013.
- [4] P. Ögren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *Automatic Control, IEEE Transactions on*, vol. 49, no. 8, pp. 1292–1302, 2004.
- [5] F. Borrelli and T. Keviczky, "Distributed lqr design for identical dynamically decoupled systems," *Automatic Control, IEEE Transactions on*, vol. 53, no. 8, pp. 1901–1912, 2008.
- [6] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Transactions on Automatic Control*, vol. 61, 2016.
- [7] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 5, no. 4, pp. 649–664, 2011.
- [8] K. Zhou and S. I. Roumeliotis, "Multirobot active target tracking with combinations of relative observations," *Robotics, IEEE Transactions on*, vol. 27, no. 4, pp. 678–695, 2011.
- [9] R. Graham and J. Cortés, "Adaptive information collection by robotic sensor networks for spatial estimation," *Automatic Control, IEEE Transactions on*, vol. 57, no. 6, pp. 1404–1419, 2012.
- [10] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *Automatic Control, IEEE Transactions on*, vol. 54, no. 1, pp. 48–61, 2009.
- [11] D. Jakovetic, J. Xavier, and J. M. Moura, "Fast distributed gradient methods," *Automatic Control, IEEE Transactions on*, vol. 59, no. 5, pp. 1131–1146, 2014.
- [12] M. G. Rabbat, R. D. Nowak, J. Bucklew *et al.*, "Generalized consensus computation in networked systems with erasure links," in *Signal Processing Advances in Wireless Communications, 2005 IEEE 6th Workshop on*. IEEE, 2005, pp. 1088–1092.
- [13] G. Stephanopoulos and A. W. Westerberg, "The use of hestenes' method of multipliers to resolve dual gaps in engineering system optimization," *Journal of Optimization Theory and Applications*, vol. 15, no. 3, pp. 285–309, 1975.
- [14] D. Jakovetic, J. M. Moura, and J. Xavier, "Linear convergence rate of a class of distributed augmented lagrangian algorithms," *Automatic Control, IEEE Transactions on*, vol. 60, no. 4, pp. 922–936, 2015.
- [15] N. Chatzipanagiotis, D. Datcheva, and M. M. Zavlanos, "An augmented lagrangian method for distributed optimization," *Mathematical Programming*, pp. 1–30, 2013.
- [16] D. Jakovetic, J. Xavier, and J. M. Moura, "Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication," *Signal Processing, IEEE Transactions on*, vol. 59, no. 8, pp. 3889–3902, 2011.
- [17] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network newton-part i: Algorithm and convergence," *arXiv preprint arXiv:1504.06017*, 2015.
- [18] S. Kar and J. M. Moura, "Gossip and distributed kalman filtering: weak consensus under weak detectability," *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1766–1784, 2011.
- [19] P. Braca, S. Marano, V. Matta, and P. Willett, "Asymptotic optimality of running consensus in testing binary hypotheses," *Signal Processing, IEEE Transactions on*, vol. 58, no. 2, pp. 814–825, 2010.
- [20] D. Bajović, D. Jakovetić, J. Xavier, B. Sinopoli, and J. M. Moura, "Distributed detection via gaussian running consensus: Large deviations asymptotic analysis," *Signal Processing, IEEE Transactions on*, vol. 59, no. 9, pp. 4381–4396, 2011.
- [21] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed kalman filtering and smoothing," *Automatic Control, IEEE Transactions on*, vol. 55, no. 9, pp. 2069–2084, 2010.
- [22] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1185–1197, 2014.
- [23] A. Simonetto, A. Mokhtari, A. Koppel, G. Leus, and A. Ribeiro, "A decentralized prediction-correction method for networked time-varying convex optimization," in *Proceedings of the 6th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (Cancun, Mexico)*, 2015.
- [24] A. Simonetto, A. Koppel, A. Mokhtari, G. Leus, and A. Ribeiro, "A quasi-newton prediction-correction method for decentralized dynamic convex optimization," in *Control Conference (ECC), 2016 European*, 2016.
- [25] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "A decentralized second-order method with exact linear convergence rate for consensus optimization," *arXiv preprint arXiv:1602.00596*, 2016.
- [26] M. R. Hestenes, "Multiplier and gradient methods," *Journal of optimization theory and applications*, vol. 4, no. 5, pp. 303–320, 1969.
- [27] D. P. Bertsekas, *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.
- [28] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.